直流电子负载的应用及举例
操作说明书

目录

一．简介……3

1.1 程序实例概述……3

二．电源测试……3

2.1 电源拉载的瞬时(动态)特性……4

2.2 负载调节率……5

2.3 电流限制……6

三．电池测试……7

3.1 电池放电曲线……7

3.2 电池的内部电阻……8

四．直流负载的性能测试……10

4.1 转换速率……10

五．直流负载其他的应用……11

5.1 作为电压表使用……11

5.2 作为电路断路开关……12

六．程序实例……12

6.1 电池放电测试……12

6.2 作为电压表时的读数……15

6.3 作为电路断路开关……15

七．参考资料……16

一．简介

随着现代的科技持续的发展，市场上对质量测试仪器的需求也在增加，这是因为需要更好和更精确的测量来适应新生的科技。对于现代的大多数电子应用产品来说，使用可信赖和节能高效的电源成为关键。正是由于这个原因，拥有一个可以准确的测出可以给其他电器比如电动交通工具、电脑电源供应器、3G 手机甚至标准的家用电池等等供电的电源的特性的测试仪器是十分重要的。直流可编程负载就是这样的一个仪器，它可以帮助我们测试各种各样的设置，配置，方案和方法。编写这个操作说明手册的目的是为了提供一个在普遍范围内的直流负载的使用方法。一些为了测试电源特性的标准的特性测试将会详细描述。并且，在后面的章节会讲述怎样测定直流负载自身的一些方法。还提供了一些关于电子负载的应用和安装信息，以协助进行各种测试设置和数据测量。作为一个手册，所有使用 ITECH 直流负载 IT8512 作为不同设置和测试配置的一部分，它们的应用和例子程序都在后面的章节里。大多数应用可以用同样的方式在其他品牌的直流负载上进行模拟，但不包括例子程序，他们是特别为 IT8500 系列直流负载所编写的程序，对 IT8512 系列有全兼容性，但是不支持其他品牌的直流负载。一般来说，任何使用 ITECH 仪器做的测量都是特有的，但是不被保证。

1.1 程序实例概述

在这本操作手册的一些章节里，提供了很多程序实例来简化和加强测试方法同时显示出了 IT8500 系列直流负载可以通过 RS-232 接口直接用电脑远程控制的优点。除非特别声明，否则所有的例子程序都是用 python 编程语言来编写的。同时需要在电脑中安装 Python 库和 IT8500 负载的 COM 口库以便正确的进行通讯。要获得这些信息以及指导安装说明请直接访问下面的地址：
http://www.bkprecision.com/products/categories/sub_categories/models/?model=8500#software.
不幸的是这些软件都只能在 windows 操作平台上运行。然而由于 Python 良好的跨平台性能，一些用户可能有可以在其他平台上运行的库（不是必须要通过 com 口来通讯）。要获得详细的信息请仔细的阅读这些库的说明书。

虽然这些软件和例子程序都是用 Python 来编写的，使用其他编程语言的用户可以轻易的使用程序包中提供的 COM 口库用自己选择的语言来编写相似的程序，现在和 COM 口库配合使用良好的是 VB 和 VC#。其他的编程语言在给出正确的命令和相应的修改程序后应该也可以良好的用于 com 口的通讯。

二．电源测试

在不断的尝试和验证设计的过程中，对稳定电源的需求随着不断发展的科技在逐步增加。尤其在最新的电子设备的测试中，准确而精密的电源成为必需品。对于传统的电源，一些参数会限制影响电源的一些基本的性能，尤其是拉载动态特性，负载调节率和电流限制等等。在接下来的章节里，这些特性都会被作为例子来描述出为了测试和验证而进行的配置、结构以及所需要的设备。每个例子都会用 IT8512 作为配置中的一部分。

声明:下列章节中包括的普遍配置中一些设置会因为测试环境的不同而改变。如果一些细节被忽略，结果就可能跟下面章节中描述的结果完全不同或者不是特别准确。

2.1 电源拉载瞬时(动态)特性

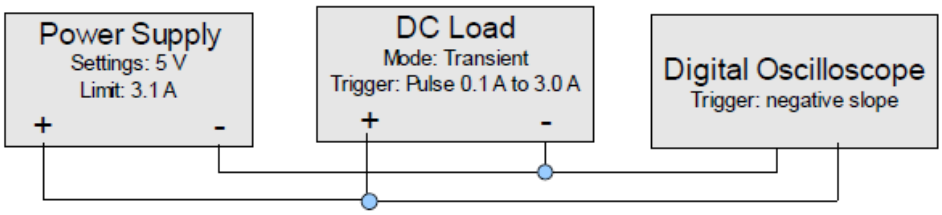

图 1. 电源拉载动态特性测试设备图

在电源测试中有一个很重要的因素是一个可以测量电源拉载动态特性的测量仪器，它可以描绘出电源在拉载电流阶跃变化时自身稳定的能力。要得到这个特性，就需要对阶跃变化时上升和下降的时间进行测试。一般来说，这种类型的测试需要电子负载的上升和下降时间必须大约比电源的快 5 倍。下面这个例子便是测量 IT6322 的拉载动态特性.

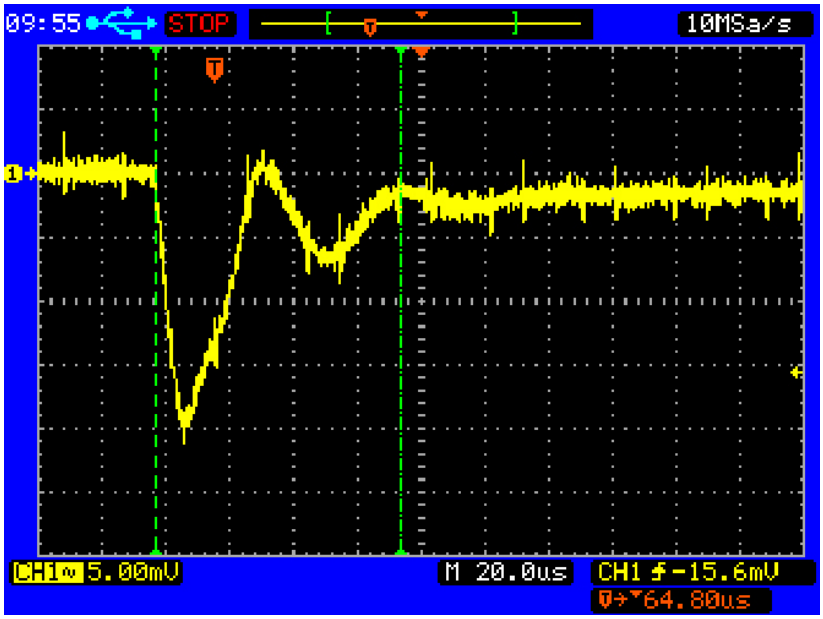

图 2. IT6322 拉载动态特性

目的：得到电源在拉载有一个阶跃变化时稳定自身输出电压所需要的时间。

方法：在电源上连接一个电子负载(IT8512 直流负载)。用电子负载的动态模式来为电源提供一个拉载的阶跃变化。

设备：将 IT6322 输出电压设置为 5V,最大电流设为 3A。用数字示波器 B&K 2542 来观察电源的输出电压，范围是交流耦合，并且为负跳变触发。

直流负载设置：直流负载被设定为动态模式，1.5A 和 3A，触发方式选择脉冲触发，意思是当脉冲来的时候负载从 1.5A 跳变至 3A，然后又跳回 1.5A。

结果：从图 2 中可以观察到当负载从 1.5A 跳变至 3A 时的上升时间就可以得到电源的拉载动态特性，输出电压下降了大约 1.6mV.

2.2 负载调节率

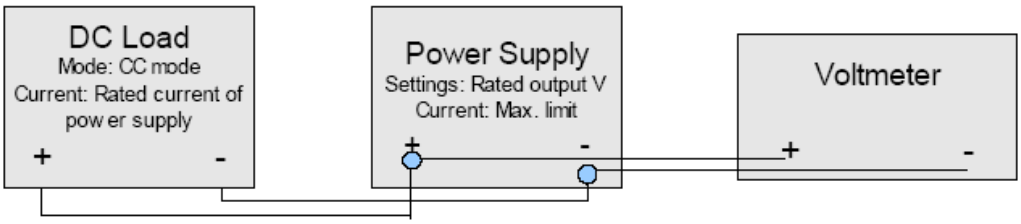

图 3. 负载调节率测试设备图

负载调节率是测试电源特性时的另一个重要的因素。这项测试需要将电源设为额定电压输出。测试是根据测电源在连接一个电子负载时电流从 0 变化至额定电流时输出电压的变化，额定电流根据测试所用的仪器的不同而不同。本测试的目的是确认电源在连接额定电流不同负载时稳定自身输出电压的能力和准确度。注意在测试前必须确认所用的电子负载能承受电源的最大额定电流和电压。

目的：测试电源从拉载电流为 0 一直到满载电流的情况下输出电压的变化情况。

方法：用一个直流电压表测量直流电子负载使电源电流从 0 一直到满载时的电压变化。

设备：从概念上来说这个测试很简单，但是实际用电压表测量电压变化时要求十分**严格**。必须将电压表连接到电源输出端子的线同电源连接负载的线分开。没有做到这点的话就会测量到电线上的电阻，测量所得的负载调节率和电源的输出电阻会偏高。如图 3 所示将直流电子负载和电压表并列的连接在电源的输出端子上。

直流负载设置：以 IT8512 为例，按 I-set 键将电子负载设置为定电流(CC)模式(您可能需要按两下这个按钮),将电子负载的电流设置为电源的额定电流。

测试步骤：

1. 打开电源并设置到输出额定电压，确认电流调节旋钮在输出最大的位置上(如果有调节旋钮的话)。
2. 记录下此时测量到的电源输出电压，记为 V_0.
3. 按下直流负载的 I-set 键将电流值设定为电源的额定电流，按 On/Off 键打开电子负载的输入。
4. 记录下此时测量到的电源输出电压，记为 V.
5. 按 On/Off 键关闭直流负载的输入。
6. 关闭电源。

按如下公式算出电源的负载调节率(百分比形式)，

$$100\frac{V_0-V}{V_0} \qquad (1)$$

同样也可以根据公式 $\frac{(V_0-V)}{i}$ 算出电源的输出电阻，其中 i 是测试中电流的下降值。

高质量的电源的输出电阻应该很小，大概在 1MΩ 或者更小。这意味着所测的 V_0 和 V 的值需要足够多的位数去避免减法时的有效位缺失。例如一个 30V，额定电流 3A，输出电阻为 1MΩ 的电源，在高量程(100V)时测量 30V 需要负载调节率有两位有效数字，于是就需要一个 6 位数的电压表。

2.3 电流限制

在电源的定电流模式时一般都有一个预设最大的输出电流的限制。电流限制测试包括了定义电源状态和电流调节率的测量。这些测量结果可以通过电压-电流曲线看出，刻画出电源是如何还有何时从 CV 模式进入 CC 模式。理想情况下的电压-电流曲线图如图 4 所示。

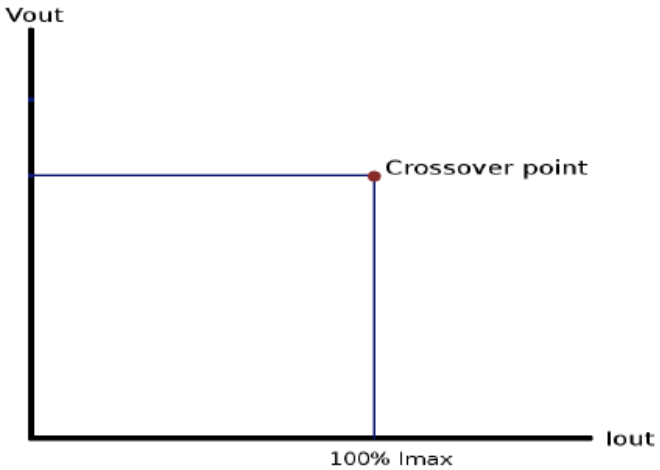

图 4. 电压-电流图

对于 CC/CV 模式电源来说，典型的电流限制特性如图 4 所示，在交叉点附近曲率非常小。测量电源的电流限制特性是个非常好的习惯，因为可以帮我们在使用时保护它。没有一定程度的电流调节率，电流可能会过功率工作甚至损害其他的设备。下面详述了以电源 IT6122 为例测试电流限制时的设备。

目的：测定待测电源的限制电流。

方法：使用一台电源，直流电子负载和一台电脑来分析电压-电流特性曲线。

设备：将电源和电子负载并列连接。将电源设定为额定电压输出，预设一个电流。当电源的输出打开时直流负载应该显示的为电源的额定电压。逐渐增加负载的拉载电流同时观察电源的输出电压的变化。当拉载电流逐步增加到接近电源的预设电流限制时，观察在转折点之前，转折点和转折后的图像，在转折点处电源由 CV 模式转为 CC 模式。

直流负载设置：以 IT8512 为例，按 I-set 键将直流电子负载设置为 CC 模式，你必须按两下，打开直流负载的输出。

结果：以电源 IT6122(额定电压为 32V,额定电流为 3A)为例,用一台直流负载来测试，观察在转折点前后的电压变化，表 1 中可以看到测量的数据。

Power supply measurements (readings from power supply)		
Voltage	Current	Mode
32.0001 V	0.00024 A	CV
31.9999 V	2.65178 A	CV
31.9999 V	2.99467 A	CV
0.1569 V	2.99999 A	CC

表 1. IT 6122 测量数据

这跟图 4 中的电压-电流曲线一致，从数据中可以看出电压在到达转折点后迅速的下降，此

时的转折点在 2.99467A. 在到达转折点以后电源切换到 CC 模式，电压保持在 0.1596V, 电流限制在 2.99999A, 这种表现对于 CC/CV 模式电源来说是非常理想的。还有其他类型的电源有不同类型的电流限制，例如反馈电流限制，CV 模式的电源(没有 CC 模式)等等。他们的电压-电流图像会和 CV/CC 模式电源有明显的差异，所以我们建议在使用电源前对它的电流限制特性进行测试。

三． 电池测试

关于存储，更新和使用能量的形式和标准都在不断的发展。对电子产品供电的新方法已经被研发出来而且比几十年前的方法更为复杂。对于现代社会对节能有效的能源的持续性的需要，工程师们现在已经把目光放在一些实际应用的能源比如燃料电池、超级电容和光电能源上。由于设计这些能源的过程十分的复杂，拥有一台可以观察到这些能源精密细节变化的可编程的测试仪器是十分有用的。多数情况下一台直流负载是非常有用的，因为它可以编程并且可以对电源(比如电池)进行放电测试。这个章节会对如何实现电池测试和内阻测试进行一个基本的概述。

3.1 电池放电曲线

当设计和测试为设备供电的电池时，通常十分被重视的是能量的效率和寿命。正是因为这样，标准的特性测试里包含了放电曲线分析，由它可以得出电池的特性。通过观察这些曲线可以测得电池的寿命，估算出电池的能效。一些直流负载提供了测量电池放电特性的功能，其中电池的容量用 AH(安培*小时)来表示，在预设的一个电压下进行放电测试。IT8500 系列直流电子负载均提供这样一个有用的内置的功能，为测试提供方便。电池放电功能同样可以用本手册在“程序实例”一章中提供的例子程序进行测试。另外，在对程序进行微小的改动之后，可能会实现更好的控制，允许测试在其他的型号和设备上进行。下面所描述的例子是对“AA”型号电池进行放电测试，可以用类似的设备配置对其他型号的电池进行测试。

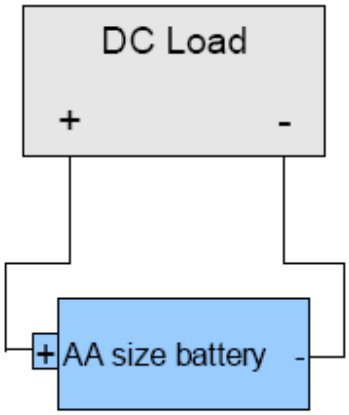

图 5. 电池放电设备图

设备：将电池连接到直流负载上，你可以将电线焊接在电池上或者用一个电池座固定电池。

直流负载设置：我们通过编程来控制负载，但是在进行测试之前要确认手动设置了电压和电流的范围。如果电流相当的大，建议使用直流负载的电压远端测量功能。

结果：如图 6 所示是以对 AA 碱性电池为例的部分放电曲线图，连接方法见图 5。用于运行这个程序的命令行见图 7。对于这个命令，必须要用通讯口 com 5 来进行通讯，并且设定负载为定电流模式(I=0.5A), 关断电压设置为 0.5V。数据文件的名称是 aa_test，并且“78deg F”是文件中每行末尾都有的一个字符串。

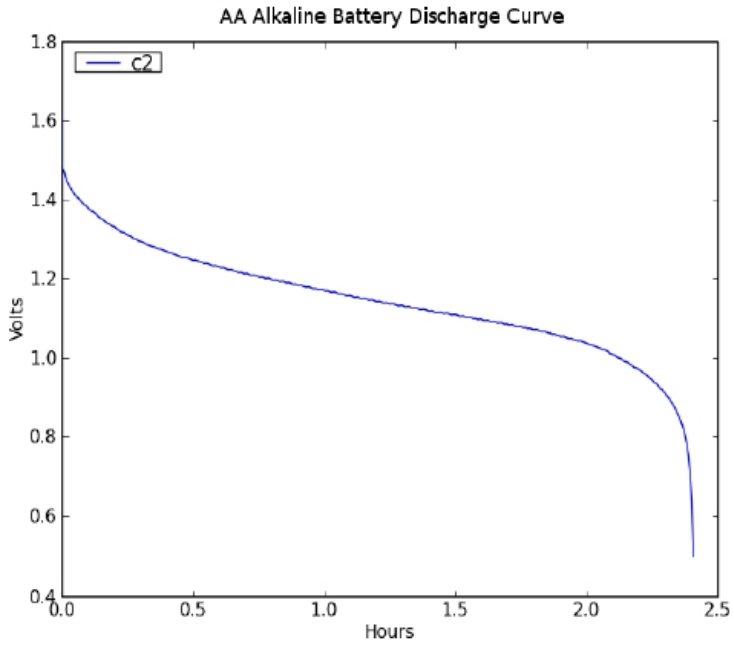

图 6.AA 碱性电池放电曲线图

```
python battery.py 5 cc 0.5 0.5 aa_test 78deg F
```

图 7. 运行程序的命令行

注意：如果您需要更多关于编程命令和如何进行测试的细节，请在本操作说明后面的章节程序实例中“电池放电测试”一节寻找用法说明和所提供的 python 程序脚本实例。

3.2 电池的内部电阻

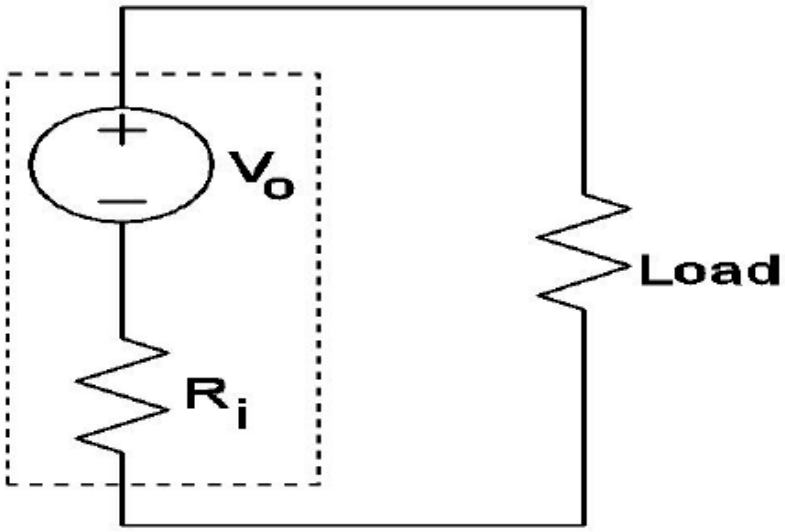

图 8. 电池内阻测量的电路模型

直流负载是测试电池内阻的一个有效的工具，如图 8 中所示，电池一般被等效模拟为如图中虚线框中所示形式。其中电源 V_0 被认为是理想的电压源并且为电池断路时候的电压值。内阻

R_i 是模拟理想情况下有电流时电池内部的阻值。如果电池可以被模拟为一个电压源和一个电阻，那么 V_0 和 R_i 就包含在等效电路图里。对于碱性电池来说，内阻随着电池的化学能的用尽而上升，随着温度的升高而降低。

如果 V_L 是通过负载的电流，那么我们有：

$$V_L = V_o + iR_i \qquad (2)$$

$$R_i = \frac{|V_L - V_o|}{i} \qquad (3)$$

可得到：

这给我们提供了一个用直流负载来估算内阻的方法，无论是通过手动调节或者通过编程来控制。

对于小电池来说，可以用编程来模拟是一个很好的优势。测试可以快速的进行并且在电池上引起的损耗小。可以用一个示波器，一个毫秒级别的电流脉冲来进行内阻的测量。下面列举了一个测量“D”型号电池内阻的例子。注意这只是一种可以测量电池内阻的方法，还有很多种没有在这里提到。

方法：在两种不同电流拉载下测量电池的电压。IT8500 系列直流电子负载的远端量测功能可以有效的避免由于电线上的压降引起的误差。测试的方法如参考[1]中所示：

提供一个稳定的 5mA 电流，测量此时的电池电压 V_0；

提供一个稳定的 505mA 电流，测量此时的电池电压 V。

$$R_i = \frac{V_o - V}{0.5} \qquad (4)$$

内阻的计算公式为：

设备：将电池连接在直流负载上，用一个电池座或者必要时将电线焊接在电池上，如果焊接的话，注意不要让电池过热以防止电池破裂。尽可能接近的将电线连接在电池的两端，并且连接电子负载后面的远端测量端子上（注意确保极性连接正确）。打开负载，在“DC Load Setup”下进行配置。确定电池电压稳定后，记录下起始电压 V_0，然后触发动态电流并且立刻观察显示的电压，记录下在两秒内显示的最低电压作为 V，然后用方程(4)计算出内阻。测试设备图如图 9 所示。

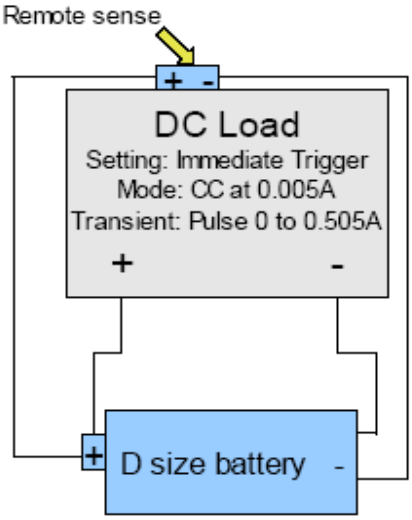

图 9. 电池内阻测量的测试设备图

直流负载设置：1.打开负载远端量测功能。

2.选择触发方式为 immediate。

3.将负载设为定电流模式，I=0.005A.

4.设置动态模式为从 0A (0.5ms)到 0.505A (2000ms)，脉冲触发(pluse)。

结果：用图 9 所示设备来进行测试，待测电池的稳定电压(V_0)是 1.469V.当有 0.505A 的拉载电流时，电压 V 为 1.415V，使用公式(4)可计算出内阻为 0.16Ω。

电池是一直复杂的非线性电化学装置，两种通用的描述电池特性的参数就是电池的直流内阻和交流阻抗。在过去，电池的交流特性对于直流操作时并不十分重要，然而，现代数字化的电气设备会从电源拉载一个急剧的电流，例如让手机发送信号。在一个 1kW 的汽车音响系统中，电流可以超过 100 A，并且电池、电容以及杂散电感等等的交流特性都可能处于主导的地位，直流负载在帮助测试这些系统的动态特性时十分有用。

如图 8 所示简单的例子中将电池看成一个理想的电压源和一个电阻串联在一起。一些更为复杂的等效模型也可以用来进行测试。运用像 IT8500 系列直流电子负载可以在一定程度上刻画出电池的特性，同时可映射出他们的应用方向和建立一个适当的模型设计。

四．直流负载的性能测试

正如我们需要一台在不同测试环境下都能精密测量的电源一样，我们同样需要一台强壮的，在特定要求和测试环境下表现优良的负载。通常会用于直流负载不同特性的测试例如触发延迟、开关翻转时间和转换速率等。在后面的章节中，会特别的介绍 IT8513C 直流负载的转换速率的测量。

声明：下列章节中包括的普遍配置中一些设置会因为测试环境的不同而改变。如果一些细节被忽略，结果就可能跟下面章节中描述的结果完全不同或者不是特别准确。

4.1 转换速率

直流负载的转换速率特性表示了直流负载在电流于不同范围内动态变化时拉电流的速度。一般来说，低电流范围时(比如 0 到 0.5 A)动态变化的转换速率会明显的比高电流范围时(比如 30 to 70 A)动态变化的速率小。通常情况下，测量转换速率的方法为在电流转换的最大时间内取一段时间来观察(如图 10 所示)，从 10%开始测量到 90%结束。转换速率可以在最陡的部分被测量出来。如图所示测得的时间可以用来计算转换速率。因此，转换速率可以很简单的由(额定最大电流-0A)/T 得到，其中 T 为最大时间的 10%到 90%. 电流为负载的额定最大电流，下面的例子表明了如何测量 IT8513C 直流负载的转换速率。

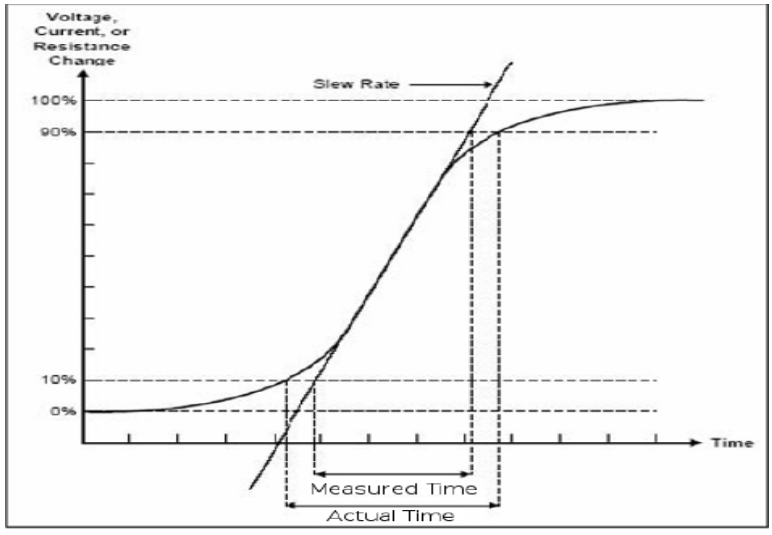

图 10. 转换速率测量图

测量方法：将直流负载设定在动态模式，电流设定为在 0A 和额定的最大电流间的转换。在示波器上观察电流的动态变化。

设备：将三台 B&K 1796 大电流电源并联。因为每台电源可以最大输出 50A 的电流，所有可以产生足够大的电流供 IT8513C 直流负载(最大电流范围为 0-120A)拉电流使用。一些额定值为(5 W 0.22 Ω)的分流电阻并联连接在 BNC 转接头适配器上。也可以用一个大的分流器来简化设备并且降低热消耗产生的温度。因为要拉载比较大的电流，所以有一个很重要的提示就是在将几个电源连接时以及连接电源和负载时要使用比较粗的导线。将电源、分流电阻和电子负载等设备如图 11 连接在一起。将 BNC 转接头适配器连接在 B&K 2542 示波器上。按照下面介绍的设置直流负载和示波器。

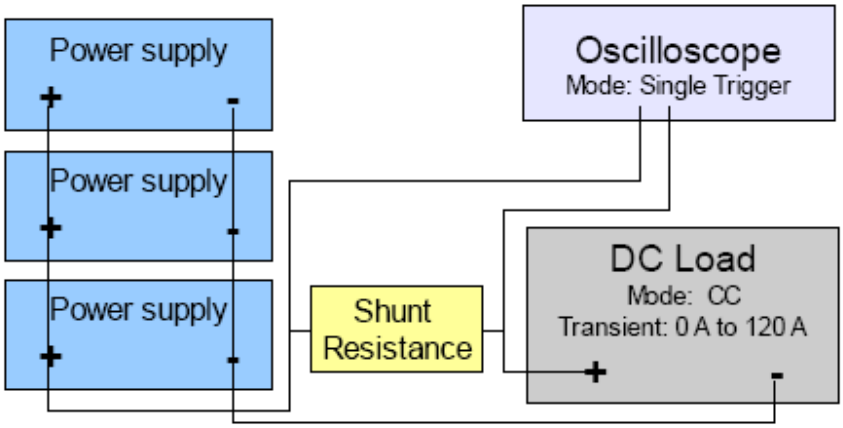

图 11. 转换速率测量设备图

直流负载设置：1.设置负载的动态模式为从 0A(0.5ms)到 120A(1000ms)，脉冲触发(pluse)。

2.将负载设置为 CC(恒流)模式。

3.将输入打开一小会儿，当示波器设置好了以后将输入关掉。

示波器设置：1.将示波器设为 single run

2.调整触发电平直到找到波形

3.调整水平和垂直方向上的量程重新触发一次

4.重复第 2 步和第 3 步直至找到最好的波形

结果：在示波器上观察波形并与图 10 进行对比，用光标在 10%和 90%处标记。观察曲线最陡的地方。可以通过两条线之间的差距进行大约的估算。因为本实验中动态模式为从 0A 到 120A，转换率可以通过在时间变化内电流的变化来计算出。变化值可以由屏幕上的光标差值得到，对于 IT 8513C 直流电子负载来说，转换率大约为 1 A/μs。

五．直流负载其他的应用

直流负载对于各种各样的测试和应用都十分有用，有时候它甚至可以在不同的测试中充当不同的测试仪器，一个很典型的例子就是充当电压表，其他的应用在本章中会提到作为电路短路开关来使用。直流电子负载可以作为电路短路开关来监控电路中的电流以来保护电路因为过流或者过功率而损坏。

5.1 作为电压表使用

在任何的设计或者测试中，都需要使用电压表来实现不同的功能，它是现在测试中最为有用的仪器之一。正是因为这样，当周围没有电压表的时候，如果拥有一台可以读取电压值的直流电子负载那会十分的方便。简单起见，在手册最后“程序实例”一章的电压表一节中提供了一个程序实例，演示如何用 IT8500 直流电子负载进行简单的测量电压和读取速度。这种负载在

没有过压的情况下有毫伏级别的分辨率。请参考那一章节来获取关于设置和程控的详细信息。

5.2 作为电路断路开关

这一章节里所讲述的是如何用电子直流负载来作为电路断路开关来使用。在程控的状态下，电子负载可以在测量到实际的无论电流、电压或者功率值比负载的预设值大的情况时断开电路。这是作为断路开关的基本用法，由于程序控制时有一定的延迟，在对反应速度要求很高是不推荐这样使用。这个程序的实例可以在后面“程序实例”一章的电路断路开关一节中找到，同时还提供了使用和操作的具体细节。

设备：将电子负载串联在电路中，注意确认远程量测功能关闭，运行控制程序来控制电流，门槛电流在程序中设置。

注意：本电子负载不会像一个纯电阻一样工作，在作为电路短路开关的时候可能很难检测到在 mA 电流范围内的小电流。对于 IT 8500 系列直流电子负载来说，有一个大约 0.1V 的开启电压，因此可能会跳过设定的小电流，所以在将本仪器用来保护一个贵重的电路时请先对作为电路断路开关的性能进行测试。

六．程序实例

在这个章节中包涵了所有在前面章节中提到过的可程控的实际应用的程序源代码。如果您的电脑中没有安装python，您也可以在下面提供的B&K网站：
http://www.bkprecision.com/products/categories/sub_categories/models/?model=8500#software 中找到您需要的文件来成功的安装python库。在使用本手册中提供的任何例子程序之前，**必须确保**这个文件被正确的安装。同时必须仔细阅读每个程序例子下的说明以便进行正确的使用。

6.1 电池放电测试

这个程序的目的就是观察平常家用电池的放电特征曲线，可以在 IT 8500 系列直流电子负载对电池放电时用来进行监控记录。

操作说明：

在代码中有两个设置可以进行修改以便用于进行不同的测试。这两个设置一个是读取的间隔时间，在程序的前面定义为“interval_s”，另一个是波特率，在main()最后定义为“baudrate”。预设值分别为 interval_s = 1，baudrate = 38400。

要运行这个程序，可以使用前面提过的这条命令：

```
python battery.py 5 cc 0.5 0.5 aa_test 78deg F
```

这个命令是特定对于通讯端口5，在CC模式下，I=0.5A，关断电压为0.5V，存储数据的文件名为“aa_test.dat”，每行代码后面都加了一个额外的字符串“78deg F”。对照下面表2得到详细的介绍。

注意：这个程序不会覆盖已有的数据文件。

Run command	Python file name	Port #	Mode:	Decimal value for constant mode	Cut off voltage	Filename of recorded data	Any string to attach at end of each line
python	Name of the python program	COM port # used to connect to PC	cc: constant current cv: constant voltage cp: constant power cr: constant resistance	Any value set for the respective mode. (i.e. 0.5 in cc mode is 0.5 A, 2 in cv mode is 2 V)	Decimal value of cut off voltage to end program	Files are saved in same folder as the python program file with .dat extension	Any strings of text can be added for reference

表 2. 电池放电测试程序指令说明表

battery.py：

```
Data file:
   Header info:
      # Command line
      # Date and time for start of test
   Entry info:
      time voltage current power

   time is in seconds since start of test
   Other measurements are in V, A, and W

We talk to a DC load using COM.
"""

import sys, dcload
from win32com.client import Dispatch
from string import join
from time import time, sleep
from msvcrt import kbhit, getch
out = sys.stdout.write
nl = "\n"
interval_s = 1   # Interval between readings

def Usage():
   name = sys.argv[0]
   msg = """Usage:  %(name)s port mode value cov filename [docstring]
Test a battery with a DC load until the specified cut-off voltage is reached.
Arguments are:
   port
      COM port that the load is at.
   mode
      cc for constant current
      cv for constant voltage
      cp for constant power
      cr for constant resistance
   value
      Decimal value of desired constant mode value.
   cov
      Cut-off voltage in volts.
   filename
      Data will be stored in filename.dat.
   docstring
      Any further text is just logged to the data file.
""" % locals()
   print msg
   exit(1)

def ProcessCommandLine():
   if len(sys.argv) < 6:
```

```
    Usage()
  port = int(sys.argv[1])
  mode = sys.argv[2]
  value = float(sys.argv[3])
  cov = float(sys.argv[4])
  filename = sys.argv[5] + ".dat"
  docstring = ""
  if len(sys.argv) > 6:
    docstring = join(sys.argv[6:])
  # Check values
  assert(port >= 0)
  assert(mode in ("cc", "cv", "cp", "cr"))
  assert(value > 0)
  assert(cov > 0)
  from os.path import exists
  assert(not exists(filename))
  out("Mode     = %s\n" % mode)
  out("Value    = %s\n" % value)
  out("cov      = %s\n" % cov)
  out("filename = %s\n" % filename)
  return port, mode, value, cov, filename, docstring

def Set(task, error_message):
  if error_message:
    out("Error on task '%s':\n" % task)
    out(error_message)
    exit(1)

def SetModeAndValue(load, mode, value):
  # Set mode and value.  Also check that the maximum setting is at
  # least the value we want.
  Set("Set to mode %s" % mode, load.SetMode(mode))
  if mode == "cc":
    max_current = float(load.GetMaxCurrent())
    if max_current < value:
      out("Max current setting is less than desired value" + nl)
      exit(1)
    Set("Set to value %g" % value, load.SetCCCurrent(value))
  elif mode == "cv":
    max_voltage = float(load.GetMaxVoltage())
    if max_voltage < value:
      out("Max voltage setting is less than desired value" + nl)
      exit(1)
    Set("Set to value %g" % value, load.SetCVVoltage(value))
  elif mode == "cp":
    max_power = float(load.GetMaxPower())
    if max_power < value:
      out("Max power setting is less than desired value" + nl)
      exit(1)
    Set("Set to value %g" % value, load.SetCWPower(value))
  else:
    Set("Set to value %g" % value, load.SetCRResistance(value))

def GetCurrentVoltagePower(load):
  def get_value(f):
    value, unit = f.split()
    return float(value)
  s = load.GetInputValues()
  fields = s.split("\t")
  voltage = get_value(fields[0])
  current = get_value(fields[1])
  power   = get_value(fields[2])
  return voltage, current, power

def RunTest(log, load, mode, value, cov):
  def LogMsg(msg):
    out(msg)
    log(msg)
  load.SetRemoteControl()
  SetModeAndValue(load, mode, value)
  log("# Command line:  %s\n" % join(sys.argv[1:]))
  'log("# %s\n" % version)'
  msg = "# Test start time = " + load.TimeNow() + nl
  LogMsg(msg)
  Set("Turn load on", load.TurnLoadOn())
  voltage, current, power = GetCurrentVoltagePower(load)
  start_time = time()
  while voltage >= cov:
    voltage, current, power = GetCurrentVoltagePower(load)
    current_time = time() - start_time
    LogMsg("%9.2f %9.4f %8.4f %9.4f\n" % \
        (current_time, voltage, current, power))
    sleep(interval_s)
    if kbhit():
      c = getch()
      if c == "Q":
        break
  Set("Turn load off", load.TurnLoadOff())
  LogMsg("# Test finish time = " + load.TimeNow() + nl)
  load.SetLocalControl()

def main():
  port, mode, value, cov, filename, docstring = ProcessCommandLine()
  load = Dispatch('BKServers.DCLoad85xx')
  baudrate = 38400
  load.Initialize(port, baudrate) # Open a serial connection
  log = open(filename, "w").write
  RunTest(log, load, mode, value, cov)

main()
```

6.2 作为电压表时的读数

在这个章节里，下面所给出的例子程序用来将电子负载作为电压表使用，它可以方便的远程控制将电压读数存储到一个数据文件里。

操作说明：

在运行程序之前必须确保有两个设置必须配置正确。这两个设置一个是通讯端口和波特率。在 voltmeter.py 的代码中，它们在程序的前面被声明，预设为“port = 5” 和“baudrate = 38400”。请务必将这两个值设定的与 PC 机上的值一致。

注意：直流电子负载必须和程序代码中的波特率保持一致，端口必须和 PC 上的设置以及电子负载本身的设置相同。

voltmeter.py

```
import sys, dcload
from time import time
from win32com.client import Dispatch
from msvcrt import kbhit

port = 5
baudrate = 38400

load = Dispatch('BKServers.DCLoad85xx')
load.Initialize(port, baudrate) # Open a serial connection
load.SetRemoteControl()
of = open("fast.dat", "w")
start = time()
count = 1
while True:
   msg = "%6d %8.2f s   %s\n" % (count, time() - start, load.GetInputValues())
   of.write(msg)
   if kbhit():
      break
   count += 1
finish = time() - start
load.SetLocalControl()
of.write("\nReading rate = %.1f readings/second\n" % (count/finish))
print "\nReading rate = %.1f readings/second\n" % (count/finish)
```

6.3 作为电路断路开关

很多直流电子负载的特点是有过电流，过电压和过功率保护，正是由于这个特性，直流负载可以作为电路断路开关来使用，用来监控电路中的电流、电压和功率。一个很好的方式是用程序来说明这个应用。下面提供的一个例子是一个 python 程序用来监控 IT 8500 系列直流电子负载的电流值。

操作说明：

在运行程序之前必须确保有两个设置必须配置正确。这两个设置一个是通讯端口和波特率。在 fuse.py 的代码中，它们在程序的前面被声明，预设为“port = 5” 和“baudrate = 38400”。请务必将这两个值设定的与 PC 机上的值一致。

下面是用来运行这个程序的一条命令： `python fuse.py 5`

其中 5 是作为预设的断路值，在这个例子中为 5A。在例子程序中进行细微的改动，便可以用在监控电压或功率。

fuse.py

```
import sys, dcload
from time import time
from win32com.client import Dispatch
from msvcrt import kbhit

port = 5
baudrate = 38400

def ReadCurrent(load):
    fields = load.GetInputValues().split("\t")  # Split on tab characters
    current = float(fields[0].split()[0])  # Remove the "A"
    return current

def main():
    # Check the command line
    if len(sys.argv) != 2:
        print "Usage:  %s fuse_value_in_A" % sys.argv[0]
        exit(1)
    # Get the desired fuse value from the command line
    fuse_value_A = float(sys.argv[1])
    assert(fuse_value_A) > 0
    # Establish a load connection
    load = Dispatch("BKServers.DCLoad85xx")
    load.Initialize(port, baudrate) # Open a serial connection
    load.SetRemoteControl()
    load.SetMode("cc")  # Set to constant current mode
    load.SetCCCurrent(fuse_value_A)  # We won't exceed this value
    load.TurnLoadOn()
    current = ReadCurrent(load)
    while current < fuse_value_A:
        if kbhit(): break
        current = ReadCurrent(load)
    load.TurnLoadOff()
    load.SetLocalControl()

main()
```

七．参考资料

[1] http://data.energizer.com/PDFs/BatteryIR.pdf A technical bulletin on internal resistance.

關於**B+K-Precision** 與 艾德克斯电子**(ITECH)**

艾德克斯电子(ITECH) 为美国第四大仪器公司B+K-Precision 集团成员，多年来一直致力于电源及电源测试领域的研究，专业生产高性能自动测试系统，电源和电子负载等大功率电子测试仪器，产品以高性能及高质量广受国内外各大企业采用，产品出口遍及欧美日等三十余国家,广泛应用于各个领域。